GraphTrack: Fast and Globally Optimal Tracking in Videos

{ Brian.Amberg and Thomas.Vetter }@unibas.ch

Problem

Special effects in movies require tracks of features through scenes. Tracks are found in an interactive process. The artist marks a position, and the computer proposes a track which is then further refined by the artist.

This is a difficult problem due to three aspects.

1. Appearance changes due to lighting and pose
2. Occlusions
3. Speed: Interactive editing requires faster than framerate calculation

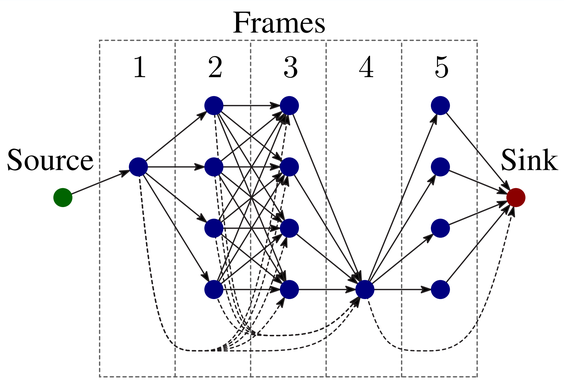
Contributions

We formulated tracking as path search in a large graph, and solve it efficiently with a modification of Dijkstra's algorithm.

The method is based on [2]. Our main contributions are

1. Efficient incorporation of a background appearance model
2. Formulation as a shortest path problem
3. (Correct) handling of occlusions
4. High-Efficiency implementation with up to 150 fps for a high resolution video

Method



The cost is interpreted as a directed acyclic graph with weights on the nodes and edges. The nodes encode candidate positions, and the edges the transition costs between candidates. Additional edges (dashed) allow occlusion transitions which skip frames.

The optimal track is found with a modification of Dijkstra's shortest path search. The search was speed up by lower bounding the cost, and lazily evaluating the accurate cost only where necessary to find the global optimum.

Background Model

We incorporate a background model, such that a click tells us not only 'this is how the landmark looks like', but also 'this is how the landmark does *not* look like' for all other patches in that frame.

The figure contrasts the per frame evidence for each candidate patch with and without a background model. Using the background model makes the correct patch probable enough, that it is chosen. But note that global reasoning over the entire path is still necessary, as the correct patch is not the most probable patch in this frame.

Results

Between one and three user clicks were needed to achieve accurate tracking for the head sequence. Note the correct handling of the occluded ear, which required only a single click.

The eye of the running giraffe required eight user interactions, of which three marked occlusions. Between one and three user clicks were needed to achieve accurate tracking for the head sequence. Note the correct handling of the occluded ear, which required only a single click.

The eye of the running giraffe required eight user interactions, of which three marked occlusions. Between one and three user clicks were needed to achieve accurate tracking for the head sequence. Note the correct handling of the occluded ear, which required only a single click.

The eye of the running giraffe required eight user interactions, of which three marked occlusions. Between one and three user clicks were needed to achieve accurate tracking for the head sequence. Note the correct handling of the occluded ear, which required only a single click.

The eye of the running giraffe required eight user interactions, of which three marked occlusions.

Speed

Speed is achieved by preprocessing the video with an adaptive filter bank as in [2]. Preprocessing was sped up significantly, but is still slower than realtime.

This encodes the video into 16 byte per pixel feature vectors. We implemented an efficient search for similar patches using the SIMD hardware of modern processors, and only evaluate the cost on these candidate patches. (Typically 200 patches per frame). Candidate search and reasoning are highly efficient resulting in an interactive system.

Note that the preprocessing is not specific to the interestpoints tracked later. A single preprocessed video can therefore be used in many annotation sessions.

References

[1] B. Amberg, T. Vetter. GraphTrack: Fast and Globally Optimal Tracking in Videos In *CVPR '11*

[2] A. Buchanan and A. Fitzgibbon. Interactive Feature Tracking using K-D Trees and Dynamic Programming. In *CVPR '06*

A Future Direction

We incorporated a background model, where a click informs us not only that 'this is how the patch looks like', but also for the rest of the frame, 'this is how the patch does not look like'.

Can we also *efficiently* use a background tracks model, allowing us to reason, 'this would be a good track, but part of it can be better explained by tracking another point'.

Contact

For more information visit us at
`http://www.pop.psu.edu`.